

BUDGET ESTIMATES MANAGEMENT



USER MANUAL

& TECHNICAL MANUAL

Table of Contents

Getting Started5
 Before you begin5
 Brief overview5
Get familiar with the interface5
 Navigation5
 Dashboard6
Budget Menu7
 Recurrent Estimates7
 Capital Estimates 12
 Reporting 13
Manage Menu 14
 Announcements 14
 Tagged Items 15
Administration Menu 16
 Budget Year 17
 This screen allows you to enter a new budget period 18
 Structure Change (REC & CAP) 18
 Migrate to New Year 19
 Export SS Files 19
 Ministries 20
 Programs 20
 Sub-Programs 21
 SOCs 23
 Categories 23
 Accounts 24
 Projects 25
 Source of Funds 25
 Sectors 25
 Tags 26
 Custom Processes 27
 Custom Reports 27
 Users & Access 29

Advanced Configuration	31
Database Overview	31
Views.....	31
Stored Procedures.....	32
Application Settings.....	32
Technologies & Resources.....	34
.Net Blazor.....	34
MudBlazor	34
Dapper	34
Microsoft SQL Server	34
Installation	35
Requirements	35
Sample Installation.....	35
APPENDIX 1 – Scripts for Main Tables.....	39
cfg_accounts	39
cfg_categories	39
cfg_ministries.....	40
cfg_programs.....	40
cfg_projects	41
cfg_sectors	42
cfg_socs	42
cfg_sofs.....	43
cfg_subprogs.....	43
cfg_tags.....	44
cfg_years	45
custom_processes.....	45
custom_reports	46
entry_status.....	47
estimates_capital.....	48
estimates_recurrent	49
notifications	51
roles.....	51
structure_change_capital.....	52

structure_change_recurrent	53
tagged_structures.....	54
users.....	55
user_access.....	56
user_roles	57
Default Data.....	58
APPENDIX 2 – Scripts for Stored Procedures.....	59
ezClearYearData	59
ezGenerateFUNDSCap.....	59
GO.....	60
ezGenerateFUNDSRec	61
ezGeneratePGWCap	62
ezGeneratePGWRec.....	62
ezGenerateUDAKCap.....	62
ezGenerateUDAKRec	63
ezMigrateData.....	63
APPENDIX 3 – Scripts for Views.....	69
vw_user_access_structure.....	69
vw_user_access	69
vw_subprogram_unsubmitted_cap	69
vw_subprogram_unsubmitted	70
vw_subprogram_submitted_cap	70
vw_subprogram_submitted	70
vw_subprogram_sub_apv_cap	70
vw_subprogram_sub_apv	70
vw_subprogram_approved_cap	71
vw_subprogram_approved.....	71

Getting Started

Before you begin

Before you begin using the software, please ensure that the software is installed and configured properly. (See **Installation Section at End of this Guide**) Once, installed, use your browser and navigate to the url.

Brief overview

The EzEstimates software provides a simple way to manage the reconciliation of accounts. Primarily it was designed to work independent as to the SmartStream system. This manual will assist you with getting familiar with the functionality of the program.

Get familiar with the interface

Navigation

The main navigation of the program is located to the left of the page.

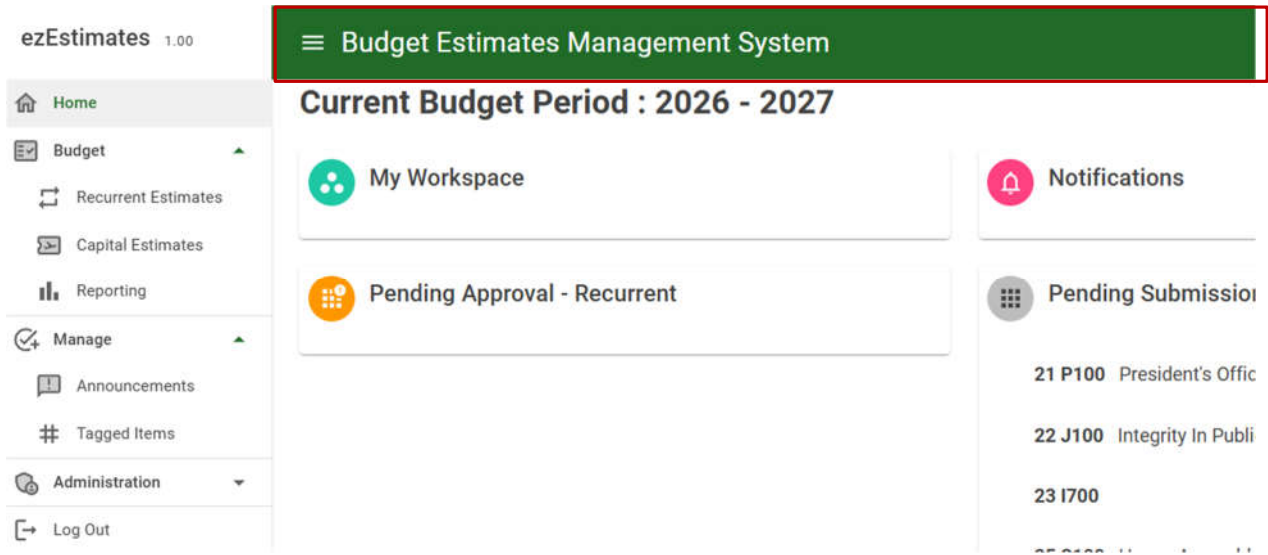


Note. The menu items that appear is based on the user security.

Menu	Role
Budget	Ministry/PSIP
Manage	Budget User
Administration	Administrator

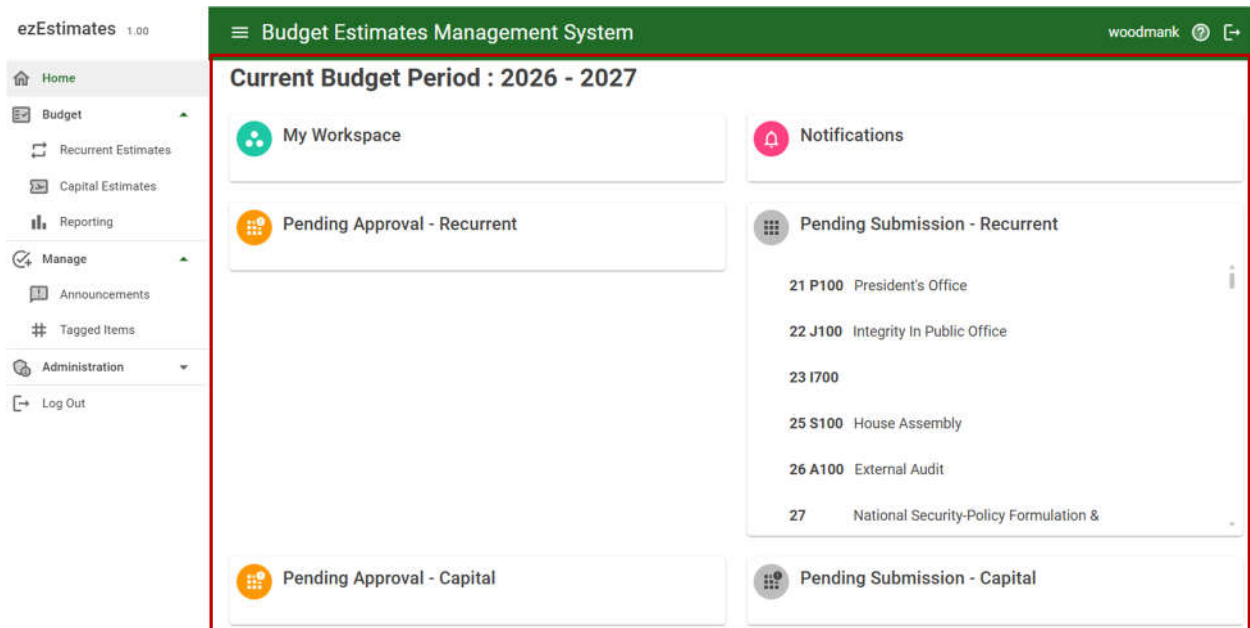
Next area of interest is the top bar.

The top bar shows the active account, the user who logged in, and also a shortcut to log out. The hamburger menu closes or opens the main navigation of the program



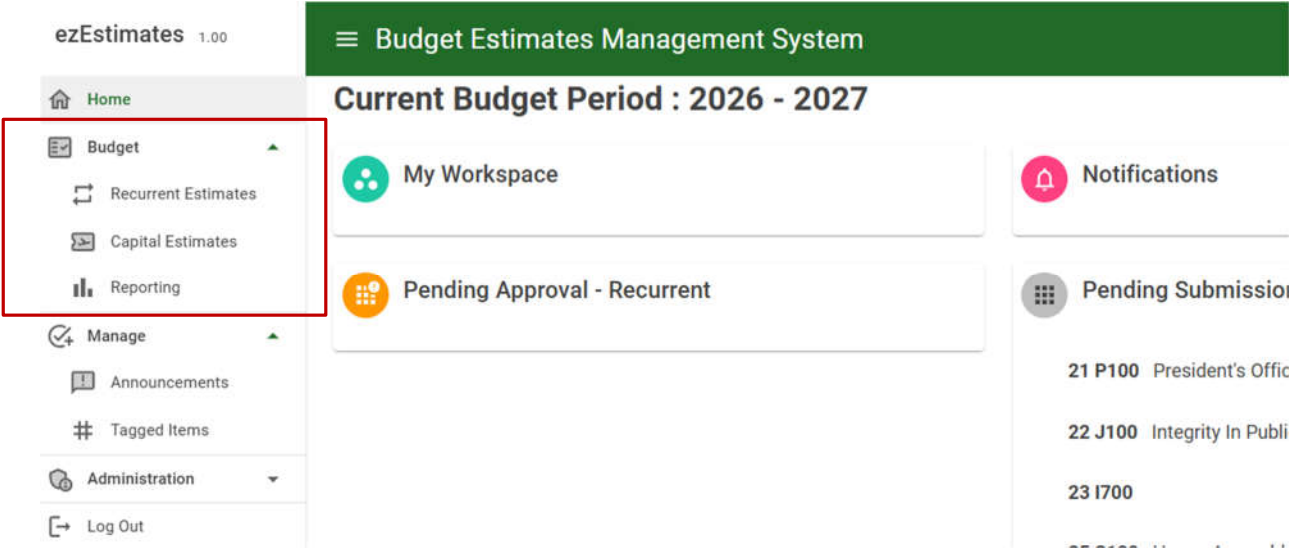
Dashboard

To Access the Dashboard click on the Home menu. The dashboard shows information on depending on the access level of the individual. They will see the programs that they usually work with, along with any notifications send by budget



Budget Menu

Let's look at the Reconciliation Menu.



There are 3 sub-menu items.

- Recurrent Estimates
- Capital Estimates
- Reporting

Recurrent Estimates

This is where majority of the work will be done. Let's familiarize ourselves with the interface.



The area highlighted allows you to select the Budget year, Change the current display to view the information by SOC or to add a new entry into the system.

Recurrent Expenditure Estimates

Budget Year: 2026

Ministry Estimates - 2026

[VIEW BY SOC](#) [ADD NEW ENTRY](#)

#	Ministry	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029
21	Office of the President	\$2,323,654	\$2,255,192	\$2,255,192	\$0
22	Integrity in Public Office	\$1,039,304	\$1,038,304	\$1,038,304	\$0
23	Public and Police Service Commission	\$903,572	\$904,594	\$912,526	\$0

This shows the summary of the estimates for the ministry. Clicking on the row takes you to more details

Icon	Function
	Shows the list of any reports that have been configured to run

Now, let's move down a level.

Recurrent Estimates - Ministry of Agriculture, Fisheries, Blue and Green Economy

2026 > (32) Ministry of Agriculture, ...

2026/2027 32 [ADD NEW ENTRY](#)

#	Program	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029
G100	Policy Formulation and Administration	\$3,235,728	\$3,235,728	\$3,235,728	\$0
G200	Agriculture Planning & Administration	\$965,244	\$965,244	\$965,244	\$0
G300	Agricultural Health, Food Safety	\$7,652,206	\$7,652,206	\$7,652,206	\$0
G400	Agricultural Development	\$7,433,634	\$7,433,634	\$7,433,636	\$0
G500	Management of Fisheries	\$2,762,384	\$2,762,384	\$2,762,384	\$0
Total		\$22,049,196	\$22,049,196	\$22,049,198	\$0

The highlighted area shows the year, where we are and also doubles as navigation.

Let's move down another level.

2026 > (32) Ministry of Agriculture, ... > (G100) Policy Formulation and ...

2026/2027 32 G100 **ADD NEW ENTRY**

#	Subprogram	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029
✓	G10 Policy Formulation and Administration	\$3,235,728	\$3,235,728	\$3,235,728	\$0
Total		\$3,235,728	\$3,235,728	\$3,235,728	\$0

Reaching this level, we are now able to take action. Clicking the down arrow reveals the ability to submit the budget for that program to budget.

2026 > (32) Ministry of Agriculture, ... > (G100) Policy Formulation and ...

2026/2027 32 G100 **ADD NEW ENTRY**

#	Subprogram	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029
⬇	G10 Policy Formulation and Administration	\$3,235,728	\$3,235,728	\$3,235,728	\$0
▶ SUBMIT					
Total		\$3,235,728	\$3,235,728	\$3,235,728	\$0

Let's go down a further.

2026 > (32) Ministry of Agriculture, ... > (G100) Policy Formulation and ... > (G10) Policy Formulation and A...

2026/2027 32 G100 G10 **ADD NEW ENTRY**

#	Account Type	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029
310	Personal Emoluments	\$1,891,096	\$1,891,096	\$1,891,096	\$0
313	Salaried Allowances	\$103,964	\$103,964	\$103,964	\$0
314	Non Salaried Allowances	\$189,058	\$189,058	\$189,058	\$0
318	Local Travel & Subsistence Allowance	\$24,960	\$24,960	\$24,960	\$0
319	International Travel & Subsistence Allowance	\$108,000	\$108,000	\$108,000	\$0
327	Training	\$2,434	\$2,434	\$2,434	\$0
332	Supplies & Materials	\$60,300	\$60,300	\$60,300	\$0
334	Communications Expenses	\$2,000	\$2,000	\$2,000	\$0
336	Operating & Maintenance Services	\$36,410	\$36,410	\$36,410	\$0
338	Rental of Assets	\$260,832	\$260,832	\$260,832	\$0
340	Professional and Consultancy Services	\$515,120	\$515,120	\$515,120	\$0
346	Subsidies (Social Assistance)	\$26,270	\$26,270	\$26,270	\$0
352	Sundry Expenses	\$3,944	\$3,944	\$3,944	\$0
452	Other Machinery and Equipment	\$11,340	\$11,340	\$11,340	\$0
Total		\$3,235,728	\$3,235,728	\$3,235,728	\$0
▶ SUBMIT					

You can see the details for the account type. You can also submit the budget from this screen.

Next, let's go a bit deeper.

2026/2027 32 G100 G10 310 **ADD NEW ENTRY**

#	Account	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029	Tags
31001	Salaries	\$1,743,078	\$1,743,078	\$1,743,078	\$0	#
31003	Overtime	\$12,000	\$12,000	\$12,000	\$0	#
31080	Social Security Contributions - Employer	\$136,018	\$136,018	\$136,018	\$0	#
Total		\$1,891,096	\$1,891,096	\$1,891,096	\$0	

Icon	Function
#	Allows you to tag accounts. Multiple tags can be assigned to one account

Clicking on the down arrow reveals the details of the figures.

2026/2027 32 G100 G10 310 **ADD NEW ENTRY**

#	Account	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029	Tags	
31001	Salaries	\$1,743,078	\$1,743,078	\$1,743,078	\$0	#	
By Law	Description	Qty	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029	
<input type="radio"/>	Junior Clerks (5)	10	237,963	237,963	237,963	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/>	Minister	2	230,356	230,356	230,356	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/>	Permanent Secretary	1	107,216	107,216	107,216	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/>	Senior Administrative Officer(1)	1	69,894	69,894	69,894	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/>	Administrative Assistant	1	49,719	49,719	49,719	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/>	Senior Executive Officer	1	49,719	49,719	49,719	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/>	Executive Officer	1	44,351	44,351	44,351	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/>	Senior Clerks (1)	1	37,449	37,449	37,449	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/>	Messenger	1	23,272	23,272	23,272	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/>	Pesticide Control Inspectors	9	21,600	21,600	21,600	0	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
ADD ROWS							

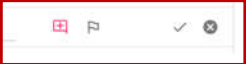
From here, you can add new rows (new information) or edit what is there.

The Actions in the grid provide the following functionality.

Icon	Functionality
	Indicates whether there is a comment on the entry or not
	Indicates where entry has been flagged and there is a comment present
	Deletes the entry

When an item is clicked, it enters **edit** mode, allowing you to edit the line.

#	Account	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029	Tags
31001	Salaries	\$1,743,078	\$1,743,078	\$1,743,078	\$0	
By Law	Description	Qty	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029
<input type="checkbox"/>	Junior Clerks (5)	10	237,963	237,963	237,963	0



Icon	Functionality
	Add a comment. The Add comment dialog box appears <div data-bbox="391 499 834 842" data-label="Image"> </div>
	Flag the item
	Save edit
	Cancel edit

For Budget Users

Once an entry has been submitted, a new icon appears to indicate that it is submitted

2026 > (21) Office of the President > (P100) Policy Formulation and ...

2026/2027 21 P100 ADD NEW ENTRY

#	Subprogram	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029
P11	Office of the President	\$2,323,654	\$2,255,192	\$2,255,192	\$0
Total		\$2,323,654	\$2,255,192	\$2,255,192	\$0

If it is approved the icon changes to indicate that

2025 > (21) Office of the President > (P100) Policy Formulation and ...

2025/2026 21 P100

#	Subprogram	Estimates 2024/2025	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028
P11	Office of the President	\$0	\$1,155,746	\$1,121,192	\$1,121,305
Total		\$0	\$1,155,746	\$1,121,192	\$1,121,305

Additional Action items for Budget Users

2026 > (21) Office of the President > (P100) Policy Formulation and ...

2026/2027 21 P100 **ADD NEW ENTRY**

#	Subprogram	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029
P11	Office of the President	\$2,323,654	\$2,255,192	\$2,255,192	\$0
Total		\$2,323,654	\$2,255,192	\$2,255,192	\$0

Buttons:

Return To Ministry – Allows the ministry to edit the items again

Adjustment - Allows the user to enter an adjustment to the Budget

Approve – Approves the budget

Run Process – Runs a custom process (eg. A custom process could be created to cut the budget by 5%)

Un-Approve – If the budget was approved, it can be reverted

Capital Estimates

All the information under **Recurrent Estimates** applies to capital estimates. The only difference is the grouping as can be seen in the following image.

Capital Expenditure Estimates

Budget Year
2026

Project Estimates - 2026

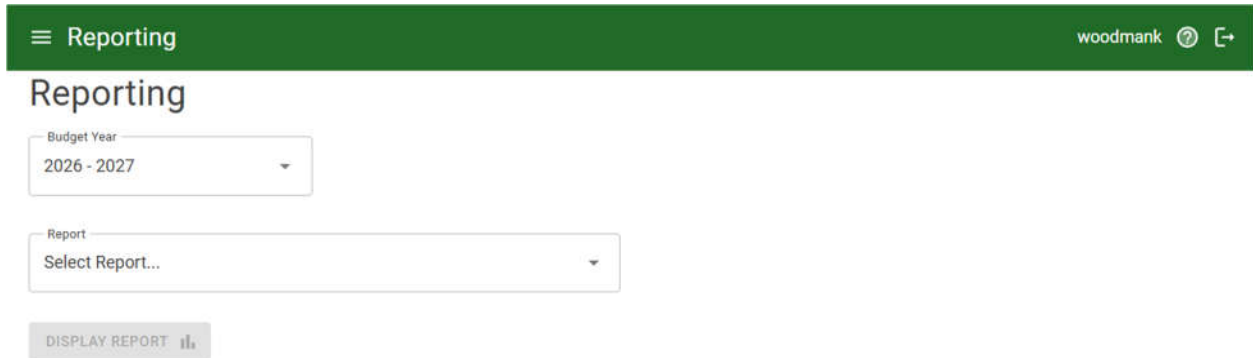
ADD NEW ENTRY

#	Project	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029
Total		\$0	\$0	\$0	\$0

Reporting

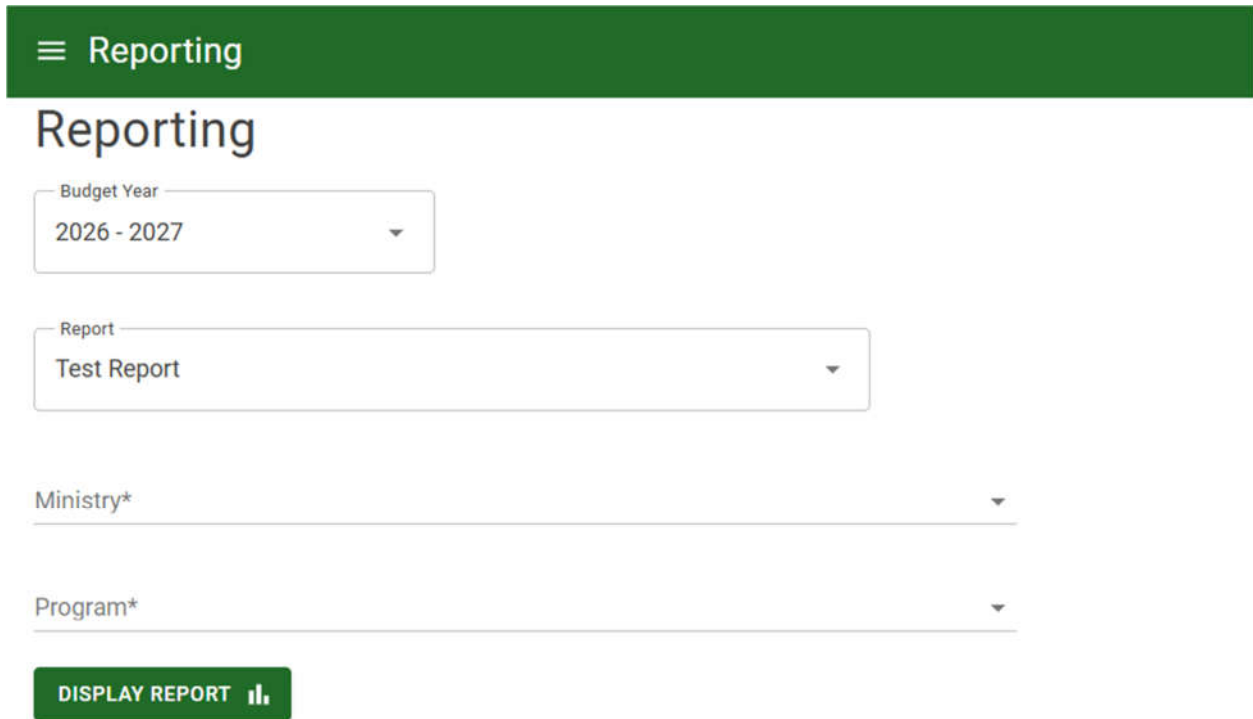
Select Reporting under the menu. This screen handles most of the reporting needs of the program.

Under the reporting section, select the report you want to run.



The screenshot shows the Reporting interface. At the top is a green navigation bar with a hamburger menu icon, the text 'Reporting', the user name 'woodmark', a help icon, and a refresh icon. Below the navigation bar, the word 'Reporting' is displayed in a large font. There are two dropdown menus: the first is labeled 'Budget Year' and shows '2026 - 2027'; the second is labeled 'Report' and shows 'Select Report...'. Below these is a grey button labeled 'DISPLAY REPORT' with a printer icon.

Once a report is selected, the options will appear so that the user can set the parameters for the report. Eg.

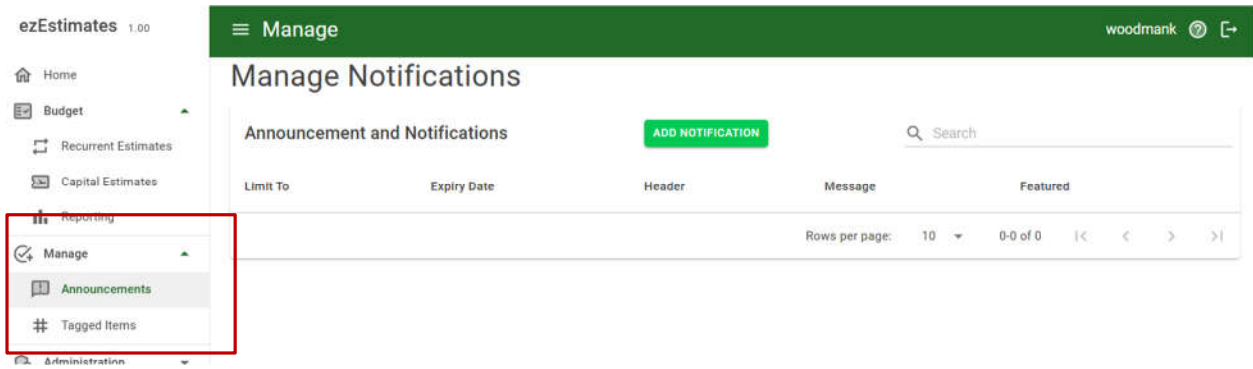


This screenshot shows the Reporting interface after a report has been selected. The 'Report' dropdown now displays 'Test Report'. Below it are two more dropdown menus: 'Ministry*' and 'Program*', both with downward arrows. At the bottom is a green button labeled 'DISPLAY REPORT' with a printer icon.

Note. All reports must be configured by Administrator. There are no built-in reports in this program.

Manage Menu

The Manage section of the program is for Budget Staff users.



Announcements

The first screen allows the user to post notifications to users of the program or groups of users.

To add a new notification, press the ADD NOTIFICATION button. The following screen will then pop-up.

A screenshot of a pop-up form titled 'Add New Notification'. The form contains the following fields and options: 'Expiry Date*' with the value '2024-11-10' and a calendar icon; 'Limit to' with a subtext 'Separate subprograms with a comma (eg. F11, F22, P10)'; 'Message Header'; 'Message*'; and two checkboxes: 'Featured' and 'Email Users'. At the bottom, there are two buttons: 'CANCEL' and 'ADD NOTIFICATION'.

Expiry date – Message will not show after expiry date


Limit to - allows the user to send messages to certain users


Featured – Stylized notifications

Email Users – send email to the users

The announcements show up on dashboard as shown below

Current Budget Period : 2026 - 2027

 My Workspace

 Notifications

Budget us around the corner
Please start to gather all necessary information

Followup
This is another announcement

Tagged Items

This screen shows all items that have a tagged attached to it.

Note. You cannot tag items from here

≡ Manage woodmark ⓘ ↗

View Tagged Items

Tagged Items Tag
Select Tag... ▾

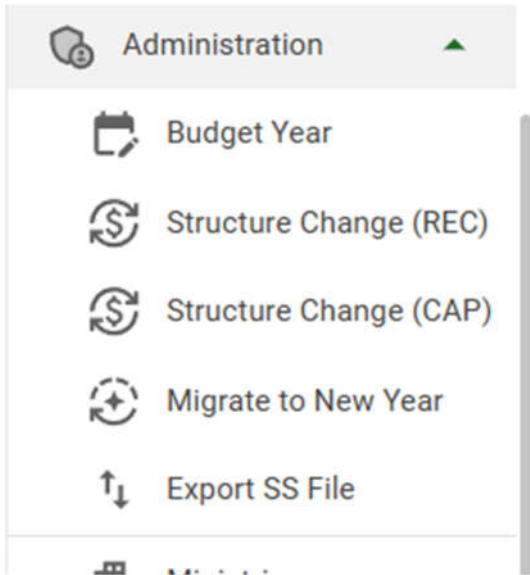
Year	Ministry	Program	Sub-Program	Project	SOF	Sector
Rows per page: 10 ▾ 0-0 of 0 < < > >						

Enter a term and press the search button. The system will return all matching transactions on both bank and ledger side irrespective of whether or not they are matched.

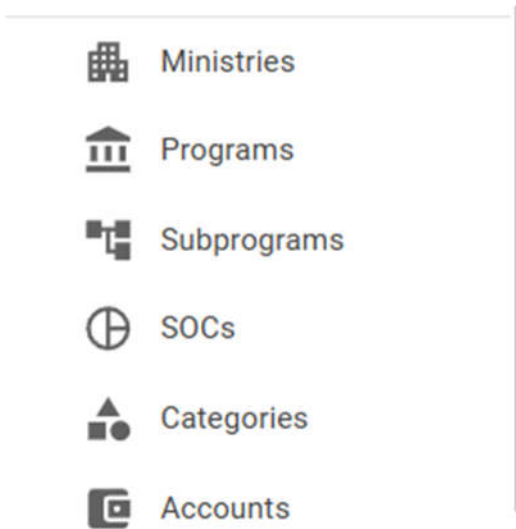
Administration Menu

Let's look at the Administration Menu.

Part 1 – Main Items for each budget year



Part 2 – Name of items



Part 3 – Names of items for capital budget

—

 Projects


 Source of Funds

 Sectors

 Tags

Part 4 – Custom actions to extend functionality of program

 Custom Processes

 Custom Reports

 Users & Access

Budget Year

Manage Budget Years

Budget Year			ADD NEW YEAR Search	
Year	Description	Year Closed		
2026	2026 - 2027	False	EDIT	DELETE
2025	2025	True	EDIT	DELETE
2024	2024	True	EDIT	DELETE
2023	2023	True	EDIT	DELETE
2022	2022	True	EDIT	DELETE
2021	2021	True	EDIT	DELETE
2020	2020	True	EDIT	DELETE
2019	2019	True	EDIT	DELETE
2018	2018	True	EDIT	DELETE

Rows per page: 10 | 1-9 of 9 | < >

This screen allows you to enter a new budget period

Structure Change (REC & CAP)

Manage Structure Changes

Recurrent Structure Changes					ADD STRUCTURE CHANGE Budget Year: 2024	
Description	Ministry	Program	Subprogram	Account		
Min Labor	27 ↳ 57	X200 ↳ OB00	X20 ↳ OB1	↳	EDIT	DELETE
Min Labor	27 ↳ 57	X200 ↳ OB00	X20 ↳ OB1	↳	EDIT	DELETE
Min Foreign Affairs	29 ↳ 53	T100 ↳ Y100	T10 ↳ Y10	↳	EDIT	DELETE
Min Foreign Affairs	29 ↳ 53	T100 ↳ Y100	T10 ↳ Y10	↳	EDIT	DELETE

Allows you to merge items from one ministry to the next

Migrate to New Year

At the start of each new year, the data from the previous year needs to be migrated. Year 1 data now becomes Year 0 data, and Year 2 data now becomes Year 1 data. The following table shows which is Year 0 and Year 1 data.

#	Ministry	Year 0	Year 1	Year 2	Year 3
		Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029
21	Office of the President	\$2,323,654	\$2,255,192	\$2,255,192	\$0
22	Integrity in Public Office	\$1,039,304	\$1,038,304	\$1,038,304	\$0

Export SS Files

Once the budget is complete, use this screen to generate the files for Smartstream.

Ministries

Administration woodmark

Manage Ministries

Ministries ADD MINISTRY Budget Year: 2026 - 2027

Ministry	Description	EDIT	DELETE
21	Office of the President	EDIT	DELETE
21	Office of the President	EDIT	DELETE
22	Integrity in Public Office	EDIT	DELETE
22	Integrity in Public Office	EDIT	DELETE
23	Public and Police Service Commission	EDIT	DELETE
23	Public and Police Service Commission	EDIT	DELETE
25	Legislature	EDIT	DELETE
25	Legislature	EDIT	DELETE
26	Audit Department	EDIT	DELETE
26	Audit Department	EDIT	DELETE

Rows per page: 10 1-10 of 44

Enter the names and codes of the ministries

Programs

Manage Programs

Programs		ADD PROGRAM	Budget Year
Program	Description		2026 - 2027
A100	Policy Formulation and Administration	EDIT DELETE	
B100	Human Resource Policy	EDIT DELETE	
B200	Facilities Management	EDIT DELETE	
CB00	Citizen Empowerment	EDIT DELETE	
D100	Policy Development & Implementation	EDIT DELETE	
D200	Socio-Economic Planning	EDIT DELETE	
D300	Physical Planning	EDIT DELETE	
D700	Energy	EDIT DELETE	
E100	Policy Formulation and Administration	EDIT DELETE	
E200	Pre-Primary and Primary Education	EDIT DELETE	

Rows per page: 10 1-10 of 103 < > >|

Enter the names and codes of the programs

Sub-Programs

Manage Subprograms

Subprograms		ADD SUBPROGRAM	Budget Year
Subprogram	Description		2026 - 2027
A10	Policy Formulation and Administration	EDIT DELETE	
B11	Policy Development & Implementation	EDIT DELETE	
B12	Resourcing and Support Services	EDIT DELETE	
B13	Training and Development	EDIT DELETE	
B14	Information Technology	EDIT DELETE	
B21	Financial Management	EDIT DELETE	
B22	Plant and Equipment	EDIT DELETE	
D70	Energy	EDIT DELETE	
E11	General Administration	EDIT DELETE	
E13	Education Planning	EDIT DELETE	

Rows per page: 10 1-10 of 223

Enter the names and codes of the sub-programs

SOCs

Manage Socs

Budget Year: 2026 - 2027

SOC	Description	EDIT	DELETE
310	Personal Emoluments	EDIT	DELETE
312	Wages (Casual Labour)	EDIT	DELETE
313	Salaried Allowances	EDIT	DELETE
314	Non Salaried Allowances	EDIT	DELETE
316	Retiring Benefits	EDIT	DELETE
318	Local Travel & Subsistence Allowance	EDIT	DELETE
319	International Travel & Subsistence Allowance	EDIT	DELETE
321	Commissions	EDIT	DELETE
323	Rewards & Incentives	EDIT	DELETE
325	Hosting & Entertainment	EDIT	DELETE

Rows per page: 10 1-10 of 33

Enter the SOC descriptions

Categories

Manage Categories

ADD CATEGORY Search

Category	Description	EDIT	DELETE
DEBT	Debt	EDIT	DELETE
NSA	Non Salary Allowance	EDIT	DELETE
OT	Overtime Services	EDIT	DELETE
PE	Personal Emoluments	EDIT	DELETE
PEN	Retiring Benefits	EDIT	DELETE
REC	Recurrent Expenditure	EDIT	DELETE
SA	Salaried Allowances	EDIT	DELETE

Rows per page: 10 1-7 of 7

Enter the category descriptions for the accounts

Accounts

Administration woodmark

Manage Accounts

Budget Year: 2026 - 2027

ADD ACCOUNT

Account	Description	Category		
31001	Salaries	PE	EDIT	DELETE
31002	Bonuses/Honoraria	PE	EDIT	DELETE
31003	Overtime	OT	EDIT	DELETE
31004	Salary Increase (Back Pay)	PE	EDIT	DELETE
31005	Social Security Contributions (Employer)	PE	EDIT	DELETE
31006	Bonus Pay	PE	EDIT	DELETE
31080	Social Security Contributions - Employer	PE	EDIT	DELETE
31081	Social Security Contributions	PE	EDIT	DELETE
31201	Wages	PE	EDIT	DELETE
31202	Bonuses/Honoraria	PE	EDIT	DELETE

Rows per page: 10 1-10 of 168

Enter the different accounts used

Projects

Administration woodmark

Manage Projects

Projects ADD PROJECT Budget Year: 2026 - 2027

Project	Description	EDIT	DELETE
C1107	BNTF6	EDIT	DELETE
P1104	2011 Population Census	EDIT	DELETE

Rows per page: 10 1-2 of 2

This screen is used to manage the projects

Source of Funds

Administration woodmark

Manage Source of Funds

Source of funds ADD SOF Budget Year: 2025

SoF	Description	EDIT	DELETE
101	Agricultural Industrial & Development Bank	EDIT	DELETE
102	Bank of Nova Scotia	EDIT	DELETE
103	Banque Francaise Commerciale	EDIT	DELETE
104	Barclays Bank PLC	EDIT	DELETE
105	Dominica Social Security	EDIT	DELETE
106	National Commercial Bank of Dominia	EDIT	DELETE
107	National Development Foundation of Dominica	EDIT	DELETE
108	National Mortgage Finance Company of Dominica Ltd	EDIT	DELETE
109	Small Projects Assistance Team	EDIT	DELETE
110	Royal Bank of Canada	EDIT	DELETE

Rows per page: 10 1-10 of 170

This screen allows you to configure the source of funds need for projects.

Sectors

Manage Sectors

Budget Year: 2026 - 2027

Sector	Description	EDIT	DELETE
101	Airport Development	EDIT	DELETE
102	Electricity Generation/Transmission	EDIT	DELETE
103	Industry & Manufacturing	EDIT	DELETE
104	Port Development	EDIT	DELETE
105	Roads and Bridges	EDIT	DELETE
106	Sea Defenses	EDIT	DELETE
107	Water/Sewerage	EDIT	DELETE
108	Drainage & Irrigation	EDIT	DELETE
201	Agriculture	EDIT	DELETE
202	Agro-Processing	EDIT	DELETE

Rows per page: 10 1-10 of 26

This screen allows you to manage the sectors

Tags

Manage Tags

ADD TAG Search

Tag	Description	EDIT	DELETE
GW	Global Warming	EDIT	DELETE

Rows per page: 10 1-1 of 1

This screen allows you to create tags that you may attach to an account.

The following is an example of the tags.

#	Account	Estimates 2025/2026	Estimates 2026/2027	Estimates 2027/2028	Estimates 2028/2029	Tags
✓	31001 Salaries	\$505,843	\$464,590	\$467,007	\$0	# GW

Custom Processes

Administration woodmark

Manage Custom Processes

Custom Process ADD CUSTOM PROCESS

Process Name	Stored Procedure	Hidden	Use Parameter	Use Year Parameter	Use Username Parameter
--------------	------------------	--------	---------------	--------------------	------------------------

Rows per page: 10 0-0 of 0

Custom processes allows you to extend the functionality of the program. This processes are *Stored Procedures* that the user creates on the database.

Add New Process

Friendly Name*

Stored Procedure Name*

Hidden

Use a parameter

Use year parameter

Use username parameter

CANCEL ADD NEW PROCESS

Note. Hidden processes do not show up in the process menu on the list

Note. If you are using the parameter, make sure the parameter name is **param**. The parameter name for year is **year**. And the parameter name for username is **username**.

Custom Reports

Manage Custom Reports

Custom Reports						ADD CUSTOM REPORT
Display Name	Report Name	URL Base Address	Report Setting	Stored Procedure		
Test Report	fff	http://google.com/	&rsCommand=Render&rs:Embed=true&rc:Toolbar=true&rc:Parameters=Collapsed			

Rows per page: 10 1-1 of 1 |< < > >|

This screen allows you to configure the reports used by the system.

When adding a report in the report dialog as shown below,

Add New Custom Report ×

Friendly Name*

Report Name*

Base Address*

URL Report Setting

Stored Procedure

Username

Ministry

Program

Subprog

SOC

Account

Project

SOF

Add To Min. List

You will be able to configure which parameters are necessary for the report. Also, you can have reports appear in the list by toggling the following options:

- Add To Min. List
- Add To Prog. List
- Add To Proj. List

Note. Reports added to the list automatically sends the corresponding data as a parameter to the report

Note. Stored procedure runs before the report is called.

Note the following parameter names for the options:

- parYear
- parUser
- parStartDate
- parEndDate
- parMinistry
- parProgram
- parSubprogram
- parSoc
- parAccount
- parProject
- parSof

Users & Access

Administration woodmark

Manage Users and Access

Users ADD NEW USER Search

id	Username	Name	Role(s)			
55	Abrahamj	Abrahamj	Ministry	ROLES	ACCESS	DELETE
56	accountantpmh	accountantpmh	Ministry	ROLES	ACCESS	DELETE
57	admin	admin	Ministry	ROLES	ACCESS	DELETE
58	admin_budgetadmin	admin_budgetadmin	Budget User	ROLES	ACCESS	DELETE

Maintain the users and their access to various features of the app.

The **ROLES** button opens the roles dialogue for the user. You may select what role you would like the user to have.

Manage User Roles for admin

Ministry User PSIP User Budget User Administrator

CANCEL SAVE

Click on the role to enable a role for a user.

The **ACCESS** button opens the dialog box for the user where you can specify which programs and projects that the user has access to.

Access for admin_dataentry

Grant admin_dataentry access to...

Access To	Description
-----------	-------------

ADD

admin_dataentry will have access to...

Search

Access To	Description	
F11	General Activities	X
F12	Macro Economic Planning & Policy Unit	X
F14	Building Maintenance Unit	X
F21	Financial Services Unit	X
F22	Citizenship by Investment Unit	X
F23	Budget, Debt & Fiscal Management	X

CANCEL SAVE

Once a valid subprogram/program/project is entered you may add access to it.

Advanced Configuration

Database Overview

The main tables are:

- cfg_accounts
- cfg_categories
- cfg_ministries
- cfg_programs
- cfg_projects
- cfg_sectors
- cfg_socs
- cfg_sofs
- cfg_subprogs
- cfg_tags
- cfg_years
- custom_processes
- custom_reports
- entry_status
- estimates_capital
- estimates_recurrent
- notifications
- roles
- structure_change_capital
- structure_change_recurrent
- tagged_structures
- user_access
- user_roles
- users

Views

The program uses the following views

- vw_subprogram_approved
- vw_subprogram_approved_cap
- vw_subprogram_sub_apv
- vw_subprogram_sub_apv_cap
- vw_subprogram_submitted

- vw_subprogram_submitted_cap
- vw_subprogram_unsubmitted
- vw_subprogram_unsubmitted_cap
- vw_user_access
- vw_user_access_structure
- vw_user_roles

Stored Procedures

The program also contains the following stored procedures:

- ezClearYearData (Clears the data for the year)
- ezGenerateFUNDSCap (Generates the smartstream file for funds for capital budget)
- ezGenerateFUNDSRec (Generates the smartstream file for funds for recurrent budget)
- ezGeneratePGWCap (Generates the smartstream file for PGW for capital budget)
- ezGeneratePGWRec (Generates the smartstream file for PGW for recurrent budget)
- ezGenerateUDAKCap (Generates the smartstream file for UDAK for capital budget)
- ezGenerateUDAKRec (Generates the smartstream file for PGW for recurrent budget)
- ezMigrateData (migrates data from one year to the next)

Application Settings

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "Default": "Data Source=WOODZONE;Initial Catalog=ezEstimates;Integrated Security=True",
    "Domain": "gocd.mofpdomgov.gov"
  },
  "ReportSettings": {
    "Server":
"http://gocdssr/ReportServer/Pages/ReportViewer.aspx?%2fGOCDBudgetAppReports%2f",
    "Settings": "&rsCommand=Render&rsEmbed=true&rc:Toolbar=true&rc:Parameters=Collapsed"
  },
  "EmailSettings": {
    "Server": "192.168.1.1",
    "Suffix": "@dominica.gov.dm",

```



```

    "Email": "smartstream@dominica.gov.dom",
    "User": "smartstream",
    "Password": "xxxx"
  },
  "BaseUrl": "/",
  "TimeOut": "1200",
  "NotifyBudget": "false",
  "NotifyBudgetSubject": "{0} submitted",
  "NotifyBudgetMessage": "The following subprogram was submitted for processing: {0}",
  "NotifyEndUser": "false",
  "NotifyEndUserSubject": "Budget not approved",
  "NotifyEndUserMessage": "The submission following submission {0} was not approved. Please review,
make the necessary changes and resubmit the budget."
}

```

Setting	Description
Default	The connection string to the database
Domain	The domain that the users authenticate with
Server	The server address for the reporting server
Email Settings	Information for the sending of emails
BaseUrl	The install location of the application. Eg. http://gov.apps/
TimeOut	Default database time out
NotifyBudget	Whether to notify the people with budget privileges that an item was submitted
NotifyBudgetSubject	Subject of email
NotifyBudgetMessage	Template of email message. (Note. Use {0} to insert the Sub-program information
NotifyEndUser	Whether to notify the people with ministry privileges that an item was approved and sent back
NotifyEndUserSubject	Subject of email
NotifyEndUserMessage	Template of email message. (Note. Use {0} to insert the Sub-program information

TECHNICAL INFORMATION

Technologies & Resources

.Net Blazor

The web program was built using the latest C# and Blazor web technologies. C# is a **modern, innovative, open-source, cross-platform** object-oriented programming language. And is used from enterprise development.

<https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor>

MudBlazor

Mudblazor is an opensource UI framework. It provides the various components which provide a consistent experience for the end user. It follows the Google Material Design system.

<https://mudblazor.com/>

Dapper

Dapper is a simple object mapper for .NET. It is used in the program to connect to the SQL Server database.

<https://dapperlib.github.io/Dapper/>

Microsoft SQL Server

Microsoft SQL Server is used to power the back end database in the application.

<https://www.microsoft.com/en-us/sql-server>

Installation

Requirements

The program runs on Microsoft Technologies. Due to .Net Core, it can run standalone if so desired. However, the supplied compile version was compiled to run on Microsoft ISS on windows sever.

Before you install on the windows server, please ensure that the .Net SDK is installed on the server.

You can download the SDK at

<https://dotnet.microsoft.com/en-us/download/dotnet>

For more information please check out the following website.

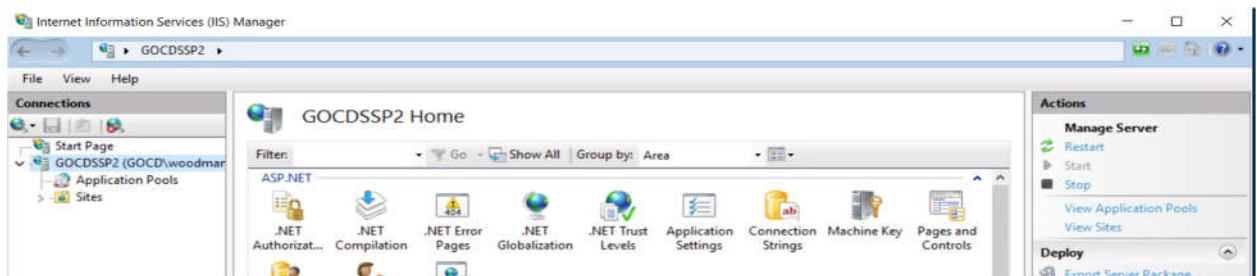
<https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/iis/?view=aspnetcore-7.0#iis-configuration>

Once the SDK is installed, you may use the WEB Deploy functionality to install the software. The software may be installed on it's own domain or part of a domain.

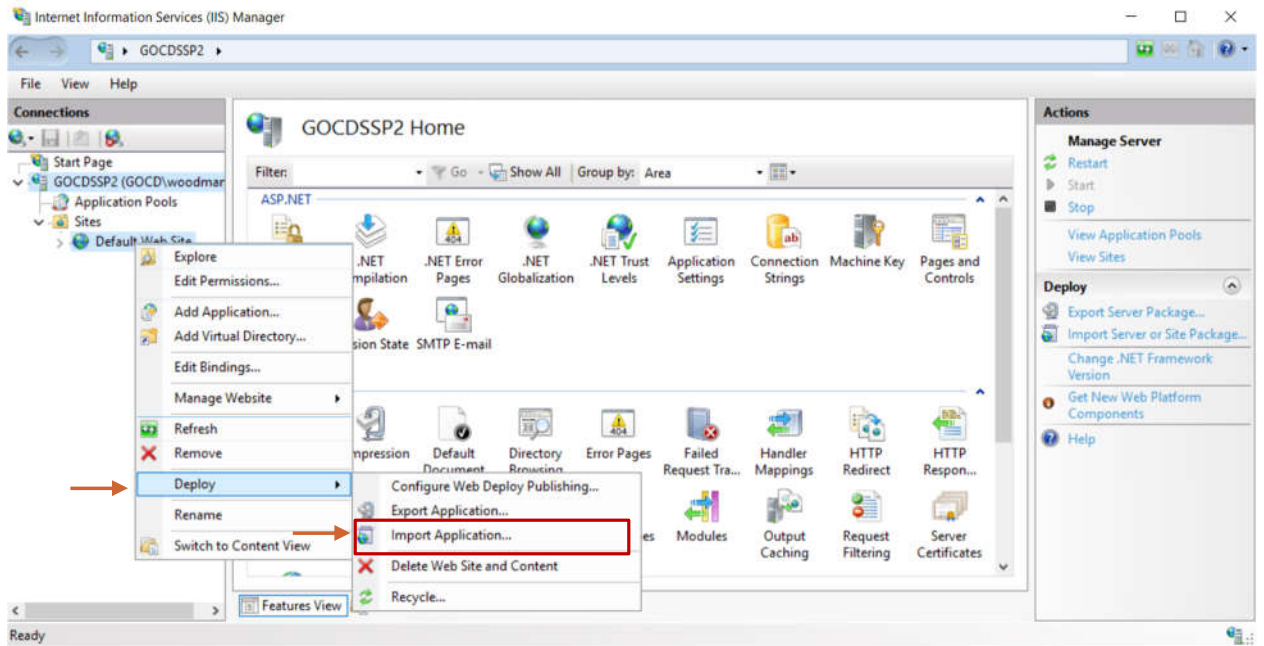
The application uses https. This provides secure communication. Therefore, a SSL certificate needs to be installed. The program will work without an SSL certificate. You may create a self signed certificate or get a one from a certificate authority.

Sample Installation

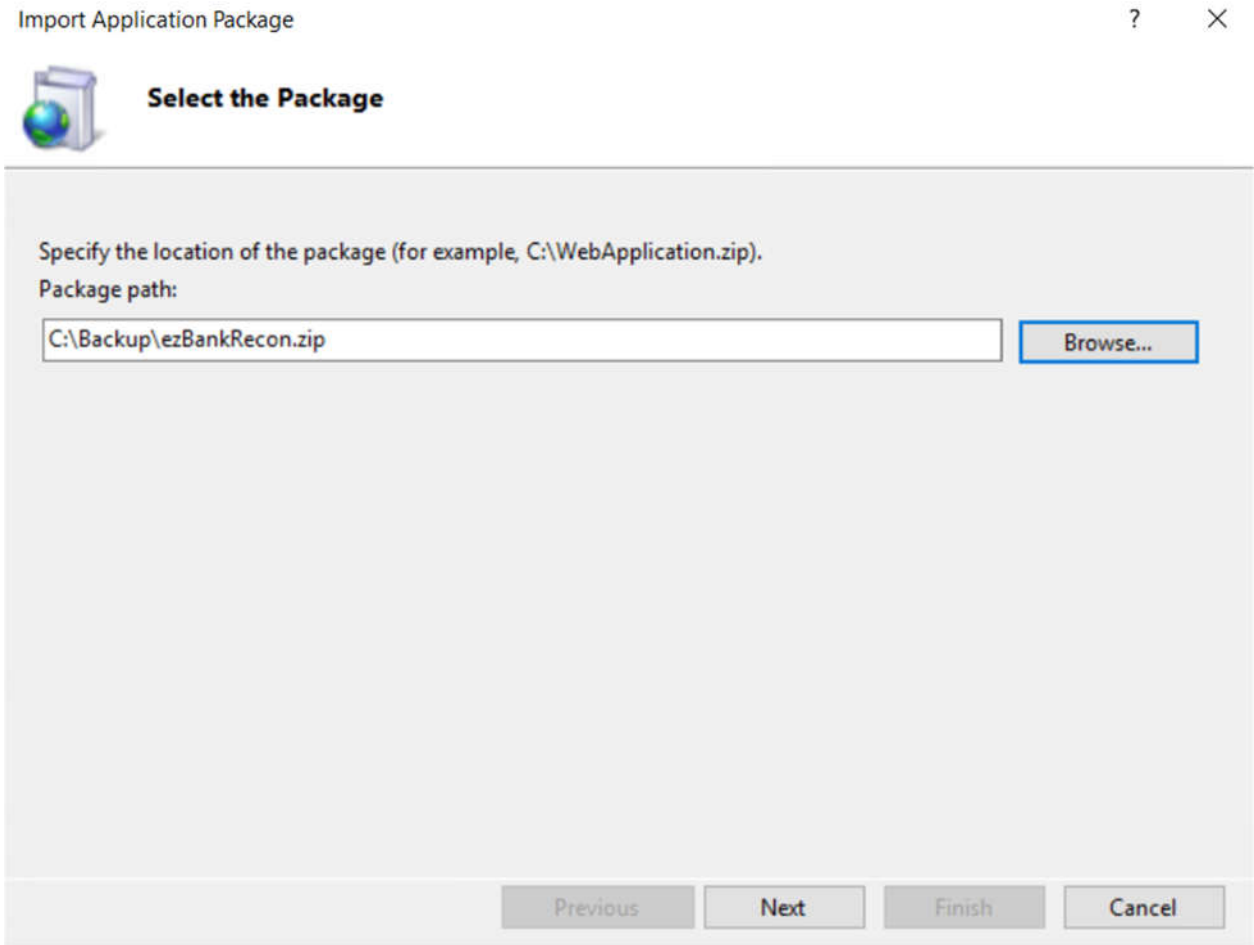
1. Open IIS



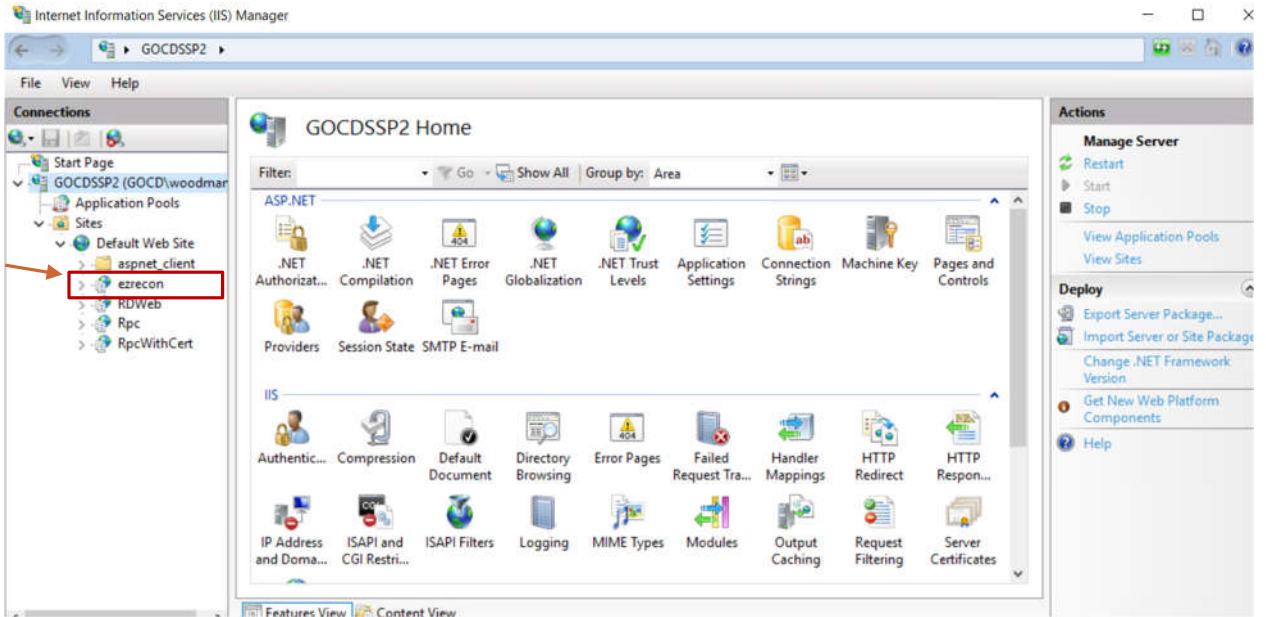
2. Select the site you would like to deploy the application to. Right Click. Select Deploy and then Import Application



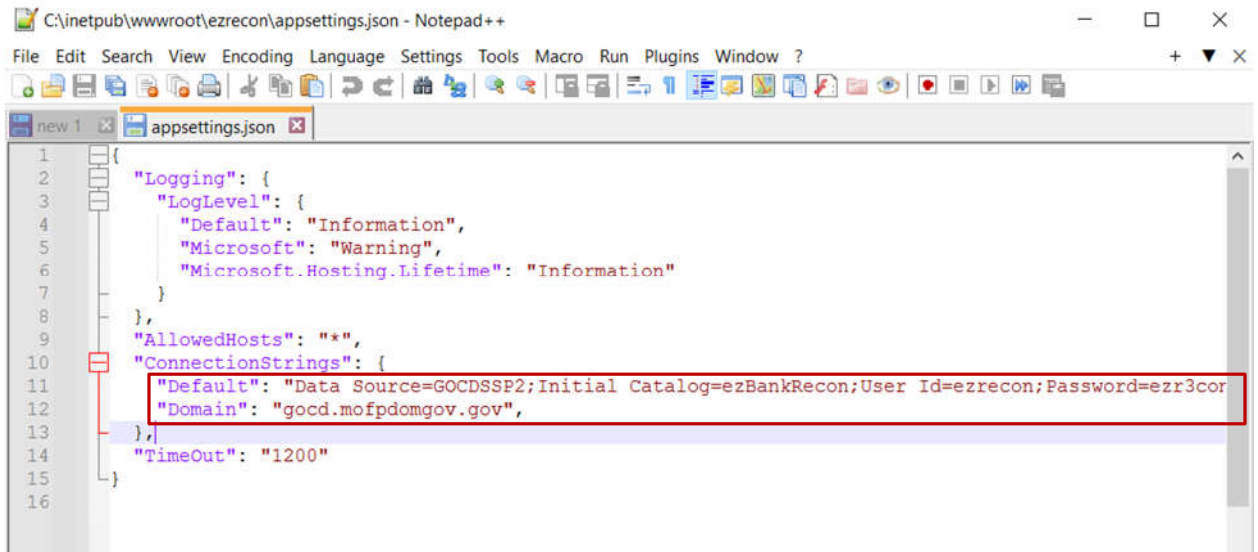
3. Follow the instructions on the wizard



4. Once complete the application should be installed



- Next, locate the config file in the inetpub, wwwroot directory. Open the folder, ezrecon and edit the appsettings.json file.



```
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft": "Warning",
6       "Microsoft.Hosting.Lifetime": "Information"
7     }
8   },
9   "AllowedHosts": "*",
10  "ConnectionStrings": {
11    "Default": "Data Source=GOCDSSP2;Initial Catalog=ezBankRecon;User Id=ezrecon;Password=ezr3cor",
12    "Domain": "gocd.mofpdomgov.gov",
13  },
14  "TimeOut": "1200"
15 }
16 }
```

- Make sure that the settings under the “ConnectionStrings” are pointing to the correct domain and correct database. And that the SQLLinkServer is configured if you have the databased linked. Save the settings and run the open the website.
- Navigate to the {url}/admin/setup to configure the database. The setup installs the database tables and creates a new admin user.



Setup New Database

Admin Name*

Admin Username*
Enter the Active Directory username

SETUP DATABASE

Enter the admin information and click SETUP DATABASE.

Once done you will get a confirmation. The program is now ready to be used.

APPENDIX 1 – Scripts for Main Tables

cfg_accounts

```
IF OBJECT_ID(N'[dbo].[cfg_accounts]', N'U') IS NULL BEGIN

    CREATE TABLE [dbo].[cfg_accounts](

        [id] [int] IDENTITY(1,1) NOT NULL,

        [year] [int] NOT NULL,

        [account] [nvarchar](50) NOT NULL,

        [descp] [nvarchar](150) NULL,

        [category] [nvarchar](5) NULL,

        CONSTRAINT [PK_account] PRIMARY KEY CLUSTERED

    (

        [id] ASC

    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,

    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON

    [PRIMARY]

    ) ON [PRIMARY]

GO

END;
```

cfg_categories

```
IF OBJECT_ID(N'[dbo].[cfg_categories]', N'U') IS NULL BEGIN

    CREATE TABLE [dbo].[cfg_categories](

        [id] [int] IDENTITY(1,1) NOT NULL,

        [category] [nvarchar](50) NOT NULL,

        [descp] [nvarchar](150) NULL,

        CONSTRAINT [PK_category] PRIMARY KEY CLUSTERED

    (

        [id] ASC

    )
```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]

) ON [PRIMARY]

GO

END;

```

cfg_ministries

```

IF OBJECT_ID(N'[dbo].[cfg_ministries]', N'U') IS NULL BEGIN

CREATE TABLE [dbo].[cfg_ministries](

[id] [int] IDENTITY(1,1) NOT NULL,

[year] [int] NOT NULL,

[ministry] [nvarchar](50) NOT NULL,

[descp] [nvarchar](150) NULL,

CONSTRAINT [PK_ministry] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]

) ON [PRIMARY]

GO

END;

```

cfg_programs

```

IF OBJECT_ID(N'[dbo].[cfg_programs]', N'U') IS NULL BEGIN

CREATE TABLE [dbo].[cfg_programs](

```



```

[id] [int] IDENTITY(1,1) NOT NULL,

[year] [int] NOT NULL,

[program] [nvarchar](50) NOT NULL,

[descp] [nvarchar](150) NULL,

CONSTRAINT [PK_program] PRIMARY KEY CLUSTERED

(

    [id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]

) ON [PRIMARY]

GO

END;

```

cfg_projects

```

IF OBJECT_ID(N'[dbo].[cfg_projects]', N'U') IS NULL BEGIN

CREATE TABLE [dbo].[cfg_projects](

    [id] [int] IDENTITY(1,1) NOT NULL,

    [year] [int] NOT NULL,

    [project] [nvarchar](50) NOT NULL,

    [descp] [nvarchar](500) NULL,

CONSTRAINT [PK_project] PRIMARY KEY CLUSTERED

(

    [id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]

) ON [PRIMARY]

```

GO

END;

cfg_sectors

```
IF OBJECT_ID(N'[dbo].[cfg_sectors]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[cfg_sectors](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [year] [int] NOT NULL,
    [sector] [nvarchar](50) NOT NULL,
    [descp] [nvarchar](150) NULL,
CONSTRAINT [PK_sector] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

END;

cfg_soc

```
IF OBJECT_ID(N'[dbo].[cfg_soc]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[cfg_soc](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [year] [int] NOT NULL,
    [soc] [nvarchar](50) NOT NULL,
```

```

        [descp] [nvarchar](150) NULL,
    CONSTRAINT [PK_soc] PRIMARY KEY CLUSTERED
    (
        [id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
    [PRIMARY]
    ) ON [PRIMARY]
GO
END;

```

cfg_sofs

```

IF OBJECT_ID(N'[dbo].[cfg_sofs]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[cfg_sofs](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [year] [int] NOT NULL,
    [sof] [nvarchar](50) NOT NULL,
    [descp] [nvarchar](150) NULL,
    CONSTRAINT [PK_sof] PRIMARY KEY CLUSTERED
    (
        [id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
    [PRIMARY]
    ) ON [PRIMARY]
GO
END;

```

cfg_subprogs

```

IF OBJECT_ID(N'[dbo].[cfg_subprogs]', N'U') IS NULL BEGIN

CREATE TABLE [dbo].[cfg_subprogs](

[id] [int] IDENTITY(1,1) NOT NULL,

[year] [int] NOT NULL,

[subprog] [nvarchar](50) NOT NULL,

[descp] [nvarchar](150) NULL,

CONSTRAINT [PK_subprogram] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,

ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON

[PRIMARY]

) ON [PRIMARY]

GO

END;

```

cfg_tags

```

IF OBJECT_ID(N'[dbo].[cfg_tags]', N'U') IS NULL BEGIN

CREATE TABLE [dbo].[cfg_tags](

[id] [int] IDENTITY(1,1) NOT NULL,

[tag] [nvarchar](50) NOT NULL,

[descp] [nvarchar](150) NULL,

CONSTRAINT [PK_tag] PRIMARY KEY CLUSTERED

(

[id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,

ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON

[PRIMARY]

) ON [PRIMARY]

GO

```

END;

cfg_years

```
IF OBJECT_ID(N'[dbo].[cfg_years]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[cfg_years](
    [year] [int] NOT NULL,
    [year_closed] [bit] NOT NULL,
    [descp] [varchar](50) NOT NULL,
    CONSTRAINT [PK__Processi__809A238A1BC99698] PRIMARY KEY CLUSTERED
(
    [year] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

END;
```

custom_processes

```
IF OBJECT_ID(N'[dbo].[custom_processes]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[custom_processes](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [process] [varchar](50) NOT NULL,
    [process_name] [varchar](50) NOT NULL,
    [hidden] [bit] NOT NULL,
    [parameter] [bit] NOT NULL,
    [include_username] [bit] NOT NULL,
```

```

    [include_year] [bit] NOT NULL,
CONSTRAINT [PK_custom_processes] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
END;

```

custom_reports

```

IF OBJECT_ID(N'[dbo].[custom_reports]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[custom_reports](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [report_name] [varchar](50) NOT NULL,
    [display_name] [varchar](50) NOT NULL,
    [base_address] [varchar](100) NOT NULL,
    [url_setting] [varchar](100) NULL,
    [stored_procedure] [varchar](50) NULL,
    [parUser] [bit] NULL,
    [parDateRange] [bit] NULL,
    [parMinistry] [bit] NULL,
    [parProgram] [bit] NULL,
    [parSubprogram] [bit] NULL,
    [parSoc] [bit] NULL,
    [parAccount] [bit] NULL,
    [parProject] [bit] NULL,
    [parSof] [bit] NULL,

```

```

[addToRecMin] [bit] NULL,
[addToRecProg] [bit] NULL,
[addToProject] [bit] NULL,
CONSTRAINT [PK_custom_reports] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

END;

```

entry_status

```

IF OBJECT_ID(N'[dbo].[entry_status]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[entry_status](
[id] [tinyint] NOT NULL,
[status_descp] [nvarchar](50) NULL,
PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

END;

```

estimates_capital

```
IF OBJECT_ID(N'[dbo].[estimates_capital]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[estimates_capital](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [year] [int] NOT NULL,
    [ministry] [char](2) NOT NULL,
    [program] [char](4) NOT NULL,
    [subprog] [char](3) NOT NULL,
    [account] [char](7) NOT NULL,
    [project] [char](5) NOT NULL,
    [sof] [char](3) NOT NULL,
    [sector] [char](3) NOT NULL,
    [lastcode] [char](4) NOT NULL,
    [quantity] [smallint] NULL,
    [year0_amount] [int] NOT NULL,
    [year1_amount] [int] NOT NULL,
    [year2_amount] [int] NOT NULL,
    [year3_amount] [int] NOT NULL,
    [is_by_law] [bit] NOT NULL,
    [comment] [nvarchar](150) NULL,
    [flagged] [bit] NOT NULL,
    [flagged_comment] [nvarchar](150) NULL,
    [modified_by] [nvarchar](50) NULL,
    [last_modified] [datetime] NULL,
    [entry_status_id] [tinyint] NOT NULL,
    [label] [nvarchar](150) NULL,
    [is_adjustment] [bit] NOT NULL,
CONSTRAINT [PK__Budget_C__3213E83FC8DF773F] PRIMARY KEY CLUSTERED
```



```

(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

END;

```

estimates_recurrent

```
IF OBJECT_ID(N'[dbo].[estimates_recurrent]', N'U') IS NULL BEGIN
```

```
CREATE TABLE [dbo].[estimates_recurrent](
```

```
    [id] [int] IDENTITY(1,1) NOT NULL,
```

```
    [year] [int] NOT NULL,
```

```
    [ministry] [char](2) NOT NULL,
```

```
    [program] [char](4) NOT NULL,
```

```
    [subprog] [char](3) NOT NULL,
```

```
    [account] [char](7) NOT NULL,
```

```
    [project] [char](5) NOT NULL,
```

```
    [sof] [char](3) NOT NULL,
```

```
    [sector] [char](3) NOT NULL,
```

```
    [lastcode] [char](4) NOT NULL,
```

```
    [quantity] [smallint] NULL,
```

```
    [year0_amount] [int] NOT NULL,
```

```
    [year1_amount] [int] NOT NULL,
```

```
    [year2_amount] [int] NOT NULL,
```

```
    [year3_amount] [int] NOT NULL,
```

```
    [is_by_law] [bit] NOT NULL,
```

```

[comment] [nvarchar](150) NULL,
[flagged] [bit] NOT NULL,
[flagged_comment] [nvarchar](150) NULL,
[modified_by] [nvarchar](50) NULL,
[last_modified] [datetime] NULL,
[entry_status_id] [tinyint] NOT NULL,
[label] [nvarchar](150) NULL,
[is_adjustment] [bit] NOT NULL,
CONSTRAINT [PK__Budget_E__3213E83FC8DF773F] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[estimates_recurrent] WITH CHECK ADD CONSTRAINT [Budget_Estimates_Status]
FOREIGN KEY([entry_status_id])
REFERENCES [dbo].[entry_status] ([id])
ON UPDATE CASCADE
ON DELETE CASCADE
GO

ALTER TABLE [dbo].[estimates_recurrent] CHECK CONSTRAINT [Budget_Estimates_Status]
GO

END;

```

notifications

```
IF OBJECT_ID(N'[dbo].[notifications]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[notifications](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [limitTo] [nvarchar](50) NULL,
    [message] [nvarchar](500) NOT NULL,
    [expiryDate] [datetime] NOT NULL,
    [dateEntered] [datetime] NOT NULL,
    [enteredby] [nvarchar](50) NOT NULL,
    [featured] [bit] NOT NULL,
    [header] [nvarchar](50) NULL,
    [dateModified] [datetime] NULL,
    [modifiedBy] [nvarchar](50) NULL,
CONSTRAINT [PK_Notifications] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

END;
```

roles

```
IF OBJECT_ID(N'[dbo].[roles]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[roles](
    [id] [int] IDENTITY(1,1) NOT NULL,
```

```

[role] [varchar](50) NOT NULL,
[role_descp] [varchar](100) NOT NULL,
CONSTRAINT [PK_ez.roles] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

END;

```

structure_change_capital

```

IF OBJECT_ID(N'[dbo].[structure_change_capital]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[structure_change_capital](
[id] [int] IDENTITY(1,1) NOT NULL,
[year] [int] NOT NULL,
[ministry] [nvarchar](50) NOT NULL,
[program] [nvarchar](50) NOT NULL,
[subprogram] [nvarchar](50) NOT NULL,
[account] [nvarchar](50) NOT NULL,
[project] [nvarchar](50) NOT NULL,
[sof] [nvarchar](50) NOT NULL,
[sector] [nvarchar](50) NOT NULL,
[to_ministry] [nvarchar](50) NOT NULL,
[to_program] [nvarchar](50) NOT NULL,
[to_subprogram] [nvarchar](50) NOT NULL,
[to_account] [nvarchar](50) NOT NULL,

```

```

[to_project] [nvarchar](50) NOT NULL,
[to_sof] [nvarchar](50) NOT NULL,
[to_sector] [nvarchar](50) NOT NULL,
[descp] [nvarchar](150) NOT NULL,
CONSTRAINT [PK_structure_change_capital] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

END;

```

structure_change_recurrent

```

IF OBJECT_ID(N'[dbo].[structure_change_recurrent]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[structure_change_recurrent](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [year] [int] NOT NULL,
    [ministry] [nvarchar](50) NOT NULL,
    [program] [nvarchar](50) NOT NULL,
    [subprogram] [nvarchar](50) NOT NULL,
    [account] [nvarchar](50) NULL,
    [to_ministry] [nvarchar](50) NOT NULL,
    [to_program] [nvarchar](50) NOT NULL,
    [to_subprogram] [nvarchar](50) NOT NULL,
    [to_account] [nvarchar](50) NULL,
    [descp] [nvarchar](150) NOT NULL,

```

```

CONSTRAINT [PK_structure_change] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

END;

```

tagged_structures

```

IF OBJECT_ID(N'[dbo].[tagged_structures]', N'U') IS NULL BEGIN
CREATE TABLE [dbo].[tagged_structures](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [year] [int] NOT NULL,
    [ministry] [char](2) NOT NULL,
    [program] [char](4) NOT NULL,
    [subprog] [char](3) NOT NULL,
    [account] [char](7) NOT NULL,
    [project] [char](5) NOT NULL,
    [sof] [char](3) NOT NULL,
    [sector] [char](3) NOT NULL,
    [tag] [int] NOT NULL,
CONSTRAINT [PK__Tags_C__3213E83FC8DF773F] PRIMARY KEY CLUSTERED
(
    [id] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON  
[PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
ALTER TABLE [dbo].[tagged_structures] WITH CHECK ADD CONSTRAINT [FK_tags_tags] FOREIGN  
KEY([tag])
```

```
REFERENCES [dbo].[cfg_tags] ([id])
```

```
GO
```

```
ALTER TABLE [dbo].[tagged_structures] CHECK CONSTRAINT [FK_tags_tags]
```

```
GO
```

```
END;
```

users

```
IF OBJECT_ID(N'[dbo].[users]', N'U') IS NULL BEGIN
```

```
CREATE TABLE [dbo].[users](
```

```
    [id] [int] IDENTITY(1,1) NOT NULL,
```

```
    [name] [varchar](50) NOT NULL,
```

```
    [username] [varchar](50) NOT NULL,
```

```
    CONSTRAINT [PK_ez.users] PRIMARY KEY CLUSTERED
```

```
(
```

```
    [id] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON  
[PRIMARY],
```

```
    CONSTRAINT [UK_users] UNIQUE NONCLUSTERED
```

```
(
```

```

        [username] ASC

    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]

    ) ON [PRIMARY]

GO

ALTER TABLE [dbo].[users] WITH CHECK ADD CONSTRAINT [FK_users_users] FOREIGN KEY([id])
REFERENCES [dbo].[users] ([id])

GO

ALTER TABLE [dbo].[users] CHECK CONSTRAINT [FK_users_users]

GO

END;

```

user_access

```

IF OBJECT_ID(N'[dbo].[user_access]', N'U') IS NULL BEGIN

    CREATE TABLE [dbo].[user_access](

        [userName] [nvarchar](50) NOT NULL,

        [accessTo] [nvarchar](5) NOT NULL,

        CONSTRAINT [PK_User_Access] PRIMARY KEY CLUSTERED

    (

        [userName] ASC,

        [accessTo] ASC

    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]

    ) ON [PRIMARY]

GO

END;

```


user_roles

```
IF OBJECT_ID(N'[dbo].[user_roles]', N'U') IS NULL BEGIN

    CREATE TABLE [dbo].[user_roles](

        [user_id] [int] NOT NULL,

        [role_id] [int] NOT NULL,

        CONSTRAINT [PK_ez.user_roles] PRIMARY KEY CLUSTERED

    (

        [user_id] ASC,

        [role_id] ASC

    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,

    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON

[PRIMARY]

    ) ON [PRIMARY]

GO

ALTER TABLE [dbo].[user_roles] WITH CHECK ADD CONSTRAINT [FK_bem.user_roles.roles] FOREIGN

KEY([role_id])

REFERENCES [dbo].[roles] ([id])

GO

ALTER TABLE [dbo].[user_roles] CHECK CONSTRAINT [FK_bem.user_roles.roles]

GO

ALTER TABLE [dbo].[user_roles] WITH CHECK ADD CONSTRAINT [FK_bem.user_roles.users] FOREIGN

KEY([user_id])

REFERENCES [dbo].[users] ([id])

GO

ALTER TABLE [dbo].[user_roles] CHECK CONSTRAINT [FK_bem.user_roles.users]
```

GO

END;

Default Data

```
INSERT INTO [dbo].[roles]([role],[role_descp])
VALUES('admin','Administrator'),('budget','Budget Staff'),('psip','PSIP Staff'),('ministry','Ministry Staff');

INSERT INTO [dbo].[users]([name],[username])
VALUES('{0}','{1}');

INSERT INTO [dbo].[user_roles]([user_id],[role_id])
VALUES(1,1);

INSERT INTO [dbo].[entry_status]([id],[status_descp])
VALUES(0,'Unsibmitted'),(1,'Submitted'),(2,'Approved');
```

APPENDIX 2 – Scripts for Stored Procedures

ezClearYearData

```
CREATE PROCEDURE [dbo].[ezClearYearData] @year int as
BEGIN
    BEGIN TRAN T1;
        DELETE FROM [estimates_recurrent]
        where year=@year;
        DELETE FROM [estimates_capital]
        where year=@year;
        DELETE FROM [cfg_accounts]
        where year=@year;
        DELETE FROM [cfg_ministries]
        where year=@year;
        DELETE FROM [cfg_programs]
        where year=@year;
        DELETE FROM [cfg_projects]
        where year=@year;
        DELETE FROM [cfg_sectors]
        where year=@year;
        DELETE FROM [cfg_soc]
        where year=@year;
        DELETE FROM [cfg_sofs]
        where year=@year;
        DELETE FROM [cfg_subprogs]
        where year=@year;
        DELETE FROM [tagged_structures]
        where year=@year;
    COMMIT TRAN T1;
END;
GO
```

ezGenerateFUNDSCap

```
CREATE PROCEDURE [dbo].[ezGenerateFUNDSCap] @proc_year int as
BEGIN
    SELECT
'ldr_entity_id,ministry,program,subprogram,soc,category,project,sof,sector,lcode,proc_year,currency,name1,name2,amt1,amount,p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14' as
Line1,0 as rowOrder
    UNION
    SELECT
'D,'+ministry+','+program+','+ec.subprog+','+SUBSTRING(ec.account,0,4)+'+',+a.category+project+','+sof+','+sector+','+lastcode+','+CONVERT(varchar,(ec.year))+',XCD,'+replace(replace(left(sp.descp, 35), ''', '-'), ',, '-')+','+replace(replace(left(s.descp, 35), ''', '-'), ',, '-')+','+CONVERT(varchar,SUM(ISNULL(year1_amount, 0)))+',0,0,0,0,0,0,0,0,0,0,0,0,0,0' as Line1, 1 as rowOrder
    FROM estimates_capital ec
    INNER JOIN cfg_soc s on SUBSTRING(account,0,4)=s.soc and s.year=ec.year
```

```

INNER JOIN cfg_accounts a on a.account=ec.account and a.year=ec.year
INNER JOIN cfg_subprogs sp on ec.subprog=sp.subprog and sp.year=ec.year
WHERE ec.year=@proc_year
GROUP BY
ministry,program,ec.subprog,SUBSTRING(ec.account,0,4),a.category,project,sof,sector,lastc
ode,ec.year,sp.descp,s.descp
HAVING SUM(ISNULL(year1_amount, 0)) <>0
UNION
SELECT
'D,'+ministry+', '+program+', '+ec.subprog+',ACCT,'+a.category+project+', '+sof+', '+sector+'
, '+lastcode+', '+CONVERT(varchar,(ec.year))+',XCD,'+replace(replace(left(sp.descp, 35),
''', '-'), ', ', '-')+ ', '+replace(replace(left(c.descp, 35), ''', '-'), ', ', '-
')+ ',0,'+CONVERT(varchar,SUM(ISNULL(year1_amount, 0)))+',0,0,0,0,0,0,0,0,0,0,0,0,0,0,0' as
Line1, 2 as rowOrder
FROM estimates_capital ec
INNER JOIN cfg_subprogs sp on ec.subprog=sp.subprog and sp.year=ec.year
INNER JOIN cfg_accounts a on a.account=ec.account and a.year=ec.year
INNER JOIN cfg_categories c on c.category=a.category
WHERE ec.year=@proc_year
GROUP BY
ministry,program,ec.subprog,a.category,project,sof,sector,lastcode,ec.year,sp.descp,c.des
cp
HAVING SUM(ISNULL(year1_amount, 0)) <>0
UNION
SELECT
'D,'+ministry+', '+ec.program+',ASP,ACCT,'+a.category+project+', '+sof+', '+sector+', '+lastc
ode+', '+CONVERT(varchar,(ec.year))+',XCD,'+replace(replace(left(p.descp, 35), ''', '-'),
', ', '-')+ ', '+replace(replace(left(c.descp, 35), ''', '-'), ', ', '-
')+ ',0,'+CONVERT(varchar,SUM(ISNULL(year1_amount, 0)))+',0,0,0,0,0,0,0,0,0,0,0,0,0,0,0' as
Line1, 3 as rowOrder
FROM estimates_capital ec
INNER JOIN cfg_programs p on ec.program=p.program and p.year=ec.year
INNER JOIN cfg_accounts a on a.account=ec.account and a.year=ec.year
INNER JOIN cfg_categories c on c.category=a.category
WHERE ec.year=@proc_year
GROUP BY
ministry,ec.program,a.category,project,sof,sector,lastcode,ec.year,p.descp,c.descp
HAVING SUM(ISNULL(year1_amount, 0)) <>0
ORDER BY rowOrder;
END;
GO

```

ezGenerateFUNDSRec

```

CREATE PROCEDURE [dbo].[ezGenerateFUNDSRec] @proc_year int as
BEGIN
    SELECT
    'ldr_entity_id,ministry,program,subprogram,soc,category,sof,sector,lcode,proc_year,curren
cy,name1,name2,amt1,amount,p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14' as Line1,0 as
rowOrder
    UNION
    SELECT
    'D,'+ministry+','+program+','+er.subprog+','+SUBSTRING(er.account,0,4)+','+a.category+', '
+sof+','+sector+','+lastcode+','+CONVERT(varchar,(er.year))+',XCD,'+replace(replace(left(
sp.descp, 35),''', '-'),',', '-')+','+replace(replace(left(s.descp, 35),''', '-'),
',', '-')+','+CONVERT(varchar,SUM(ISNULL(year1_amount,
0)))+',0,0,0,0,0,0,0,0,0,0,0,0,0,0' as Line1, 1 as rowOrder
    FROM estimates_recurrent er
    INNER JOIN cfg_soc s on SUBSTRING(account,0,4)=s.soc and s.year=er.year
    INNER JOIN cfg_accounts a on a.account=er.account and a.year=er.year
    INNER JOIN cfg_subprogs sp on er.subprog=sp.subprog and sp.year=er.year
    WHERE er.year=@proc_year
    GROUP BY
ministry,program,er.subprog,SUBSTRING(er.account,0,4),a.category,sof,sector,lastcode,er.y
ear,sp.descp,s.descp
    HAVING SUM(ISNULL(year1_amount, 0)) <>0
    UNION
    SELECT
    'D,'+ministry+','+program+','+er.subprog+',ACCT,'+a.category+','+sof+','+sector+','+lastc
ode+','+CONVERT(varchar,(er.year))+',XCD,'+replace(replace(left(sp.descp, 35),''', '-
'),',', '-')+','+replace(replace(left(c.descp, 35),''', '-'),',', '-
')+','+CONVERT(varchar,SUM(ISNULL(year1_amount, 0)))+',0,0,0,0,0,0,0,0,0,0,0,0,0,0' as
Line1, 2 as rowOrder
    FROM estimates_recurrent er
    INNER JOIN cfg_subprogs sp on er.subprog=sp.subprog and sp.year=er.year
    INNER JOIN cfg_accounts a on a.account=er.account and a.year=er.year
    INNER JOIN cfg_categories c on c.category=a.category
    WHERE er.year=@proc_year
    GROUP BY
ministry,program,er.subprog,a.category,sof,sector,lastcode,er.year,sp.descp,c.descp
    HAVING SUM(ISNULL(year1_amount, 0)) <>0
    UNION
    SELECT
    'D,'+ministry+','+er.program+',ASP,ACCT,'+a.category+','+sof+','+sector+','+lastcode+','+
CONVERT(varchar,(er.year))+',XCD,'+replace(replace(left(p.descp, 35),''', '-'),',', '-
')+','+replace(replace(left(c.descp, 35),''', '-'),',', '-
')+','+CONVERT(varchar,SUM(ISNULL(year1_amount, 0)))+',0,0,0,0,0,0,0,0,0,0,0,0,0,0' as
Line1, 3 as rowOrder
    FROM estimates_recurrent er
    INNER JOIN cfg_programs p on er.program=p.program and p.year=er.year
    INNER JOIN cfg_accounts a on a.account=er.account and a.year=er.year
    INNER JOIN cfg_categories c on c.category=a.category
    WHERE er.year=@proc_year
    GROUP BY
ministry,er.program,a.category,sof,sector,lastcode,er.year,p.descp,c.descp
    HAVING SUM(ISNULL(year1_amount, 0)) <>0
    ORDER BY rowOrder;

```

```
END;  
GO
```

ezGeneratePGWCap

```
CREATE PROCEDURE [dbo].[ezGeneratePGWCap] @proc_year int as  
BEGIN  
    SELECT  
        'D,'+ministry+', '+ec.program+', ASP,ALLOC, '+a.category+project+', '+sof+', '+sector+', '+last  
code+', '+CONVERT(varchar, (ec.year))+', XCD,,,,0, '+CONVERT(varchar, SUM(ISNULL(year1_amount,  
0)))+', 0,0,0,0,0,0,0,0,0,0,0,0,0,0' as Line1  
    FROM estimates_capital ec  
    INNER JOIN cfg_accounts a on a.account=ec.account and a.year=ec.year  
    INNER JOIN cfg_categories c on c.category=a.category  
    WHERE ec.year=@proc_year  
    GROUP BY ministry, ec.program, a.category, project, sof, sector, lastcode, ec.year  
    HAVING SUM(ISNULL(year1_amount, 0)) <>0 ;  
END;  
GO
```

ezGeneratePGWRec

```
CREATE PROCEDURE [dbo].[ezGeneratePGWRec] @proc_year int as  
BEGIN  
    SELECT  
        'D,'+ministry+', '+er.program+', ASP,ALLOC, '+a.category+project+', '+sof+', '+sector+', '+last  
code+', '+CONVERT(varchar, (er.year))+', XCD,,,,0, '+CONVERT(varchar, SUM(ISNULL(year1_amount,  
0)))+', 0,0,0,0,0,0,0,0,0,0,0,0,0,0' as Line1  
    FROM estimates_recurrent er  
    INNER JOIN cfg_accounts a on a.account=er.account and a.year=er.year  
    INNER JOIN cfg_categories c on c.category=a.category  
    WHERE er.year=@proc_year  
    GROUP BY ministry, er.program, a.category, project, sof, sector, lastcode, er.year  
    HAVING SUM(ISNULL(year1_amount, 0)) <>0 ;  
END;  
GO
```

ezGenerateUDAKCap

```
CREATE PROCEDURE [dbo].[ezGenerateUDAKCap] @proc_year int as  
BEGIN  
    SELECT  
        'ldr_entity_id,ministry,program,subprog,account,project,sof,sector,lastcode,curr_type,cur  
r_code,equiv_curr_code,proccessing_year,amt_class_type,dr_cr_code,_update_bal_code,ldr_am  
t_1,chgstamp' as Line1,0 as rowOrder  
    UNION  
    SELECT  
        'D,'+ministry+', '+program+', '+subprog+', '+account+', '+project+', '+sof+', '+sector+', '+last  
code+', TEB,XCD,XCD, '+CONVERT(varchar, (year))+', AUTHBUD,,R, '+CONVERT(varchar, SUM(ISNULL(ye  
ar1_amount, 0)))+', 0' as Line1, 1 as rowOrder
```

```

FROM estimates_capital
WHERE year=@proc_year
GROUP BY ministry,program,subprog,account,project,sof,sector,lastcode,year
HAVING SUM(ISNULL(year1_amount, 0)) <>0
UNION
SELECT
'D,'+ministry+','+program+','+subprog+','+account+','+project+','+sof+','+sector+','+last
code+',TEB,XCD,XCD,'+CONVERT(varchar,(year))+',ESTBUD,,R,'+CONVERT(varchar,SUM(ISNULL(yea
r1_amount, 0)))+',0' as Line1, 2 as rowOrder
FROM estimates_capital
WHERE year=@proc_year
GROUP BY ministry,program,subprog,account,project,sof,sector,lastcode,year
HAVING SUM(ISNULL(year1_amount, 0)) <>0
ORDER BY rowOrder;

END;
GO

```

ezGenerateUDAKRec

```

CREATE PROCEDURE [dbo].[ezGenerateUDAKRec] @proc_year int as
BEGIN

SELECT
'ldr_entity_id,ministry,program,subprog,account,project,sof,sector,lastcode,curr_type,cur
r_code,equiv_curr_code,processing_year,amt_class_type,dr_cr_code,_update_bal_code,ldr_am
t_1,chgstamp' as Line1,0 as rowOrder
UNION
SELECT
'D,'+ministry+','+program+','+subprog+','+account+','+project+','+sof+','+sector+','+last
code+',TEB,XCD,XCD,'+CONVERT(varchar,(year))+',AUTHBUD,,R,'+CONVERT(varchar,SUM(ISNULL(ye
ar1_amount, 0)))+',0' as Line1, 1 as rowOrder
FROM estimates_recurrent
WHERE year=@proc_year
GROUP BY ministry,program,subprog,account,project,sof,sector,lastcode,year
HAVING SUM(ISNULL(year1_amount, 0)) <>0
UNION
SELECT
'D,'+ministry+','+program+','+subprog+','+account+','+project+','+sof+','+sector+','+last
code+',TEB,XCD,XCD,'+CONVERT(varchar,(year))+',ESTBUD,,R,'+CONVERT(varchar,SUM(ISNULL(yea
r1_amount, 0)))+',0' as Line1, 2 as rowOrder
FROM estimates_recurrent
WHERE year=@proc_year
GROUP BY ministry,program,subprog,account,project,sof,sector,lastcode,year
HAVING SUM(ISNULL(year1_amount, 0)) <>0
ORDER BY rowOrder;

END;
GO

```

ezMigrateData

```

CREATE PROCEDURE [dbo].[ezMigrateData] @fromYear int, @toYear int as
BEGIN

```

```

BEGIN TRAN T1;
--Migrate all base table data
INSERT INTO [cfg_accounts] (year,account,descp,category)
SELECT @toYear,account,descp,category
FROM [cfg_accounts]
WHERE year=@fromYear;

INSERT INTO [cfg_ministries] (year,ministry,descp)
SELECT @toYear,ministry,descp
FROM [cfg_ministries]
WHERE year=@fromYear;

INSERT INTO [cfg_programs] (year,program,descp)
SELECT @toYear,program,descp
FROM [cfg_programs]
WHERE year=@fromYear;

INSERT INTO [cfg_projects] (year,project,descp)
SELECT @toYear,project,descp
FROM [cfg_projects]
WHERE year=@fromYear;

INSERT INTO [cfg_sectors] (year,sector,descp)
SELECT @toYear,sector,descp
FROM [cfg_sectors]
WHERE year=@fromYear;

INSERT INTO [cfg_soc] (year,soc,descp)
SELECT @toYear,soc,descp
FROM [cfg_soc]
WHERE year=@fromYear;

INSERT INTO [cfg_sofs] (year,sof,descp)
SELECT @toYear,sof,descp
FROM [cfg_sofs]
WHERE year=@fromYear;

INSERT INTO [cfg_subprogs] (year,subprog,descp)
SELECT @toYear,subprog,descp
FROM [cfg_subprogs]
WHERE year=@fromYear;

--Migrate all entries
INSERT INTO [estimates_capital](
    [year]
    ,[ministry]
    ,[program]
    ,[subprog]
    ,[account]
    ,[project]
    ,[sof]
    ,[sector]
    ,[lastcode]
    ,[quantity]
    ,[year0_amount]
    ,[year1_amount]
    ,[year2_amount]
    ,[year3_amount]

```



```

        , [is_by_law]
        , [comment]
        , [flagged]
        , [flagged_comment]
        , [entry_status_id]
        , [label]
        , [is_adjustment])
SELECT @toYear
        , [ministry]
        , [program]
        , [subprog]
        , [account]
        , [project]
        , [sof]
        , [sector]
        , [lastcode]
        , [quantity]
        , [year1_amount]
        , [year2_amount]
        , [year3_amount]
        , 0
        , [is_by_law]
        , ''
        , 0
        , ''
        , 0
        , [label]
        , 0
FROM [estimates_capital]
WHERE [is_adjustment]=0
AND year=@fromYear;

INSERT INTO [estimates_recurrent](
        [year]
        , [ministry]
        , [program]
        , [subprog]
        , [account]
        , [project]
        , [sof]
        , [sector]
        , [lastcode]
        , [quantity]
        , [year0_amount]
        , [year1_amount]
        , [year2_amount]
        , [year3_amount]
        , [is_by_law]
        , [comment]
        , [flagged]
        , [flagged_comment]
        , [entry_status_id]
        , [label]
        , [is_adjustment])
SELECT @toYear
        , [ministry]
        , [program]
        , [subprog]

```

```

, [account]
, [project]
, [sof]
, [sector]
, [lastcode]
, [quantity]
, [year1_amount]
, [year2_amount]
, [year3_amount]
, 0
, [is_by_law]
, ''
, 0
, ''
, 0
, [label]
, 0
FROM [estimates_recurrent]
WHERE [is_adjustment]=0
AND year=@fromYear;

```

--Delete any entries on the structure change

```

DELETE ec
FROM [estimates_capital] as ec
INNER JOIN [structure_change_capital] as scc
ON ec.ministry=scc.ministry and ec.program=scc.program and
ec.subprog=scc.subprogram and ec.account=scc.account and ec.project=scc.project and
ec.sof=scc.sof and ec.sector=scc.sector and scc.year=@toYear
WHERE ec.year=@toYear;

```

```

DELETE er
FROM [estimates_recurrent] as er
INNER JOIN [structure_change_recurrent] as scr
ON er.ministry=scr.ministry and er.program=scr.program and
er.subprog=scr.subprogram and er.account=scr.account and scr.year=@toYear
WHERE er.year=@toYear;

```

--Insert any entries on the structure CHANGE

```

INSERT INTO [estimates_capital](
[year]
, [ministry]
, [program]
, [subprog]
, [account]
, [project]
, [sof]
, [sector]
, [lastcode]
, [quantity]
, [year0_amount]
, [year1_amount]
, [year2_amount]
, [year3_amount]
, [is_by_law]
, [comment]
, [flagged]
, [flagged_comment]

```

```

        ,[entry_status_id]
        ,[label]
        ,[is_adjustment])
SELECT @toYear
        ,scc.[to_ministry]
        ,scc.[to_program]
        ,scc.[to_subprogram]
        ,scc.[to_account]
        ,scc.[to_project]
        ,scc.[to_sof]
        ,scc.[to_sector]
        ,[lastcode]
        ,[quantity]
        ,[year1_amount]
        ,[year2_amount]
        ,[year3_amount]
        ,0
        ,[is_by_law]
        ,''
        ,0
        ,''
        ,0
        ,[label]
        ,0
FROM estimates_capital ec
     INNER JOIN [structure_change_capital] as scc
     ON ec.ministry=scc.ministry and ec.program=scc.program and
ec.subprog=scc.subprogram and ec.account=scc.account and ec.project=scc.project and
ec.sof=scc.sof and ec.sector=scc.sector and scc.year=@toYear
WHERE ec.year= @fromYear;

```

```

INSERT INTO [estimates_recurrent](
        [year]
        ,[ministry]
        ,[program]
        ,[subprog]
        ,[account]
        ,[project]
        ,[sof]
        ,[sector]
        ,[lastcode]
        ,[quantity]
        ,[year0_amount]
        ,[year1_amount]
        ,[year2_amount]
        ,[year3_amount]
        ,[is_by_law]
        ,[comment]
        ,[flagged]
        ,[flagged_comment]
        ,[entry_status_id]
        ,[label]
        ,[is_adjustment])
SELECT @toYear
        ,scr.[to_ministry]
        ,scr.[to_program]
        ,scr.[to_subprogram]

```

```

        ,scr.[to_account]
        ,[project]
        ,[sof]
        ,[sector]
        ,[lastcode]
        ,[quantity]
        ,[year1_amount]
        ,[year2_amount]
        ,[year3_amount]
        ,0
        ,[is_by_law]
        ,''
        ,0
        ,''
        ,0
        ,[label]
        ,0
FROM estimates_recurrent er
INNER JOIN [structure_change_recurrent] as scr
ON er.ministry=scr.ministry and er.program=scr.program and
er.subprog=scr.subprogram and er.account=scr.account and scr.year=@toYear
WHERE er.year= @fromYear;
---Insert tags
INSERT INTO [tagged_structures] ([year]
        ,[ministry]
        ,[program]
        ,[subprog]
        ,[account]
        ,[project]
        ,[sof]
        ,[sector]
        ,[tag])
SELECT @toYear
        ,[ministry]
        ,[program]
        ,[subprog]
        ,[account]
        ,[project]
        ,[sof]
        ,[sector]
        ,[tag]
FROM [tagged_structures]
WHERE year= @fromYear;

COMMIT TRAN T1;
END;
GO

```

APPENDIX 3 – Scripts for Views

vw_user_access_structure

```
Create view [dbo].[vw_user_access_structure] as
SELECT DISTINCT dbo.User_Access.userName, dbo.estimated_recurrent.ministry,
dbo.estimated_recurrent.program, dbo.estimated_recurrent.subprog,
dbo.estimated_recurrent.project, year
FROM          dbo.estimated_recurrent INNER JOIN
              dbo.User_Access ON dbo.estimated_recurrent.subprog =
dbo.User_Access.accessTo INNER JOIN
              dbo.Users ON dbo.User_Access.userName = dbo.Users.userName
UNION
SELECT DISTINCT dbo.User_Access.userName, dbo.estimated_capital.ministry,
dbo.estimated_capital.program, dbo.estimated_capital.subprog,
dbo.estimated_capital.project, year
FROM          dbo.estimated_capital INNER JOIN
              dbo.User_Access ON dbo.estimated_capital.subprog =
dbo.User_Access.accessTo INNER JOIN
              dbo.Users ON dbo.User_Access.userName = dbo.Users.userName
GO
```

vw_user_access

```
Create view [dbo].[vw_user_access] as
SELECT DISTINCT dbo.User_Access.userName, ministry, m.descp as ministry_name, program,
p.descp as program_name, subprog, s.descp as subprog_name, project, j.descp as
project_name
FROM          dbo.Users
INNER JOIN    dbo.User_Access ON User_Access.userName = Users.userName
LEFT OUTER JOIN cfg_ministries m ON accessTo=ministry AND m.year =(select max(year) from
cfg_years)
LEFT OUTER JOIN cfg_programs p ON accessTo=program AND p.year =(select max(year) from
cfg_years)
LEFT OUTER JOIN cfg_subprogs s ON accessTo=subprog AND s.year =(select max(year) from
cfg_years)
LEFT OUTER JOIN cfg_projects j ON accessTo=project AND j.year =(select max(year) from
cfg_years)
GO
```

vw_subprogram_unsubmitted_cap

```
CREATE VIEW [dbo].[vw_subprogram_unsubmitted_cap]
AS
```

```

SELECT DISTINCT year, subprog, ministry, program, project, sof
FROM      dbo.estimateds_capital
WHERE     (entry_status_id = 0)
GO

```

vw_subprogram_unsubmitted

```

CREATE VIEW [dbo].[vw_subprogram_unsubmitted]
AS
SELECT DISTINCT year, subprog, ministry, program
FROM      dbo.estimateds_recurrent
WHERE     (entry_status_id = 0)
GO

```

vw_subprogram_submitted_cap

```

CREATE VIEW [dbo].[vw_subprogram_submitted_cap]
AS
SELECT DISTINCT year, subprog, ministry, program, project, sof
FROM      dbo.estimateds_capital
WHERE     (entry_status_id = 1)
GO

```

vw_subprogram_submitted

```

CREATE VIEW [dbo].[vw_subprogram_submitted]
AS
SELECT DISTINCT year, subprog, ministry, program
FROM      dbo.estimateds_recurrent
WHERE     (entry_status_id = 1)
GO

```

vw_subprogram_sub_apv_cap

```

create view [dbo].[vw_subprogram_sub_apv_cap]
as
SELECT  year, subprog, ministry, program, MAX(entry_status_id) AS status, project, sof
FROM    dbo.estimateds_capital
WHERE   (entry_status_id IN (1, 2))
GROUP BY year, project, sof, subprog, ministry, program
GO

```

vw_subprogram_sub_apv

```
CREATE VIEW [dbo].[vw_subprogram_sub_apv]
AS
SELECT year, subprog, ministry, program, MAX(entry_status_id) AS status
FROM      dbo.estimateds_recurrent
WHERE     (entry_status_id IN (1, 2))
GROUP BY year, subprog, ministry, program
GO
```

vw_subprogram_approved_cap

```
CREATE VIEW [dbo].[vw_subprogram_approved_cap]
AS
SELECT DISTINCT year, subprog, ministry, program, project, sof
FROM      dbo.estimateds_capital
WHERE     (entry_status_id = 2)
GO
```

vw_subprogram_approved

```
CREATE VIEW [dbo].[vw_subprogram_approved]
AS
SELECT DISTINCT year, subprog, ministry, program
FROM      dbo.estimateds_recurrent
WHERE     (entry_status_id = 2)
GO
```